Using the REST API

This chapter contains the following sections:

About Getting Started with APIC Examples

The steps in several examples in this guide include a parameter name. These parameter names are provided as examples for convenience and ease of your understanding, and it is not required for you to use them.

About Switch Discovery with the APIC

The APIC is a central point of automated provisioning and management for all the switches that are part of the ACI fabric. A single data center might include multiple ACI fabrics; each data center might have its own APIC cluster and Cisco Nexus 9000 Series switches that are part of the fabric. To ensure that a switch is managed only by a single APIC cluster, each switch must be registered with that specific APIC cluster that manages the fabric.

The APIC discovers new switches that are directly connected to any switch it currently manages. Each APIC instance in the cluster first discovers only the leaf switch to which it is directly connected. After the leaf switch is registered with the APIC, the APIC discovers all spine switches that are directly connected to the leaf switch.

As each spine switch is registered, that APIC discovers all the leaf switches that are connected to that spine switch. This cascaded discovery allows the APIC to discover the entire fabric topology in a few simple steps.

Switch Registration with the APIC Cluster

Note Before you begin registering a switch, make sure that all switches in the fabric are physically connected and booted in the desired configuration. For information about the installation of the chassis, see http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-installation-guides-list.html.

After a switch is registered with the APIC, the switch is part of the APIC-managed fabric inventory. With the Application Centric Infrastructure fabric (ACI fabric), the APIC is the single point of provisioning, management, and monitoring for switches in the infrastructure.

Note The infrastructure IP address range must not overlap with other IP addresses used in the ACI fabric for in-band and out-of-band networks.

Registering the Unregistered Switches Using the REST API

Note The infrastructure IP address range must not overlap with other IP addresses used in the ACI fabric for in-band and out-of-band networks.

Procedure

Register the switches.

Example:

```
POST: https://<apic-ip>/api/node/mo/uni/controller.xml
<fabricNodeIdentPol>
<fabricNodeIdentP  serial="FGE173900ZD" name="leaf1" nodeId="101"/>
<fabricNodeIdentP  serial="FGE1740010A" name="leaf2" nodeId="102"/>
<fabricNodeIdentP  serial="FGE1740010H" name="spine1" nodeId="203"/>
<fabricNodeIdentP  serial="FGE1740011B" name="spine2" nodeId="204"/>
</fabricNodeIdentPol>
```

Switch Discovery Validation and Switch Management from the APIC

After the switches are registered with the APIC, the APIC performs fabric topology discovery automatically to gain a view of the entire network and to manage all the switches in the fabric topology.

Each switch can be configured, monitored, and upgraded from the APIC without having to access the individual switches.

Validating the Registered Switches Using the REST API

Procedure

Validate switch registration using the REST API.

Example:

```
GET: https://<apic-ip>/api/node/class/topSystem.xml?

<?xml version="1.0" encoding="UTF-8"?>
<imdata>
    <topSystem address="10.0.0.1" dn="topology/pod-1/node-1/sys" fabricId="1" id="1"
name="apic1"
    oobMgmtAddr="10.30.13.44" podId="1" role="apic" serial="" state="in-service"
systemUpTime="00:00:00:02.199" .../>
    <topSystem address="10.0.0.2" dn="topology/pod-1/node-2/sys" fabricId="1" id="2"
name="apic2"
    oobMgmtAddr="10.30.13.45" podId="1" role="apic" serial="" state="in-service"
systemUpTime="00:00:00:02.199" .../>
    <topSystem address="10.0.0.3" dn="topology/pod-1/node-3/sys" fabricId="1" id="3"
name="apic3"
    oobMgmtAddr="10.30.13.46" podId="1" role="apic" serial="" state="in-service"
systemUpTime="00:00:00:02.199" .../>
    <topSystem address="10.0.98.127" dn="topology/pod-1/node-101/sys" fabricId="1" id="101"

    name="leaf1" oobMgmtAddr="0.0.0.0" podId="1" role="leaf" serial="FOX-270308"
state="in-service"
    systemUpTime="00:00:00:02.199" .../>
    <topSystem address="10.0.98.124" dn="topology/pod-1/node-102/sys" fabricId="1" id="102"

    name="leaf2" oobMgmtAddr="0.0.0.0" podId="1" role="leaf" serial="FOX-270308"
state="in-service"
    systemUpTime="00:00:00:02.199" .../>
    <topSystem address="10.0.98.125" dn="topology/pod-1/node-203/sys" fabricId="1" id="203"

    name="spine2" oobMgmtAddr="0.0.0.0" podId="1" role="spine" serial="FOX-616689"
state="in-service"
    systemUpTime="00:00:00:02.199" .../>
    <topSystem address="10.0.98.126" dn="topology/pod-1/node-204/sys" fabricId="1" id="204"

    name="spine1" oobMgmtAddr="0.0.0.0" podId="1" role="spine" serial="FOX-616689"
state="in-service"
    systemUpTime="00:00:00:02.199" .../>
</imdata>
```

Validating the Fabric Topology

After all the switches are registered with the APIC cluster, the APIC automatically discovers all the links and connectivity in the fabric and discovers the entire topology as a result.

Validating the Fabric Topology Using the REST API

Procedure

Validate the fabric topology using the REST API.

Example:

Identifiers for user reference are as follows:

- n1 = Identifier of the first node
- s1 = Slot on the first node
- p1 = Port on slot s1
- n2 = Identifier of the second node
- s2 = Slot on the second node
- p2 = Port on slot s2

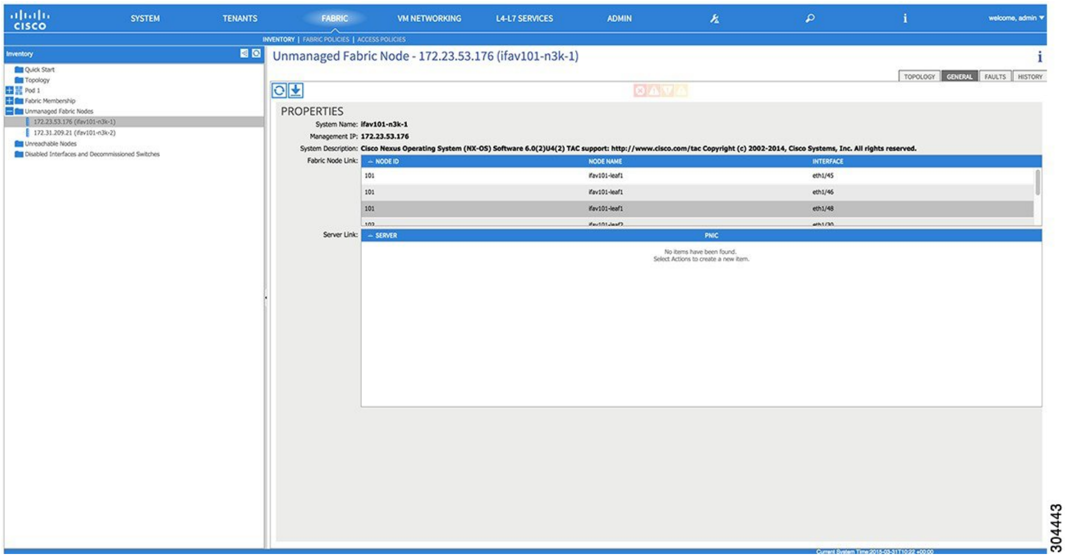
```
GET: https://<apic-ip>/api/node/class/fabricLink.xml?

<?xml version="1.0" encoding="UTF-8"?>
<imdata>
   <fabricLink dn="topology/lnkcnt-19/lnk-18-1-50-to-19-5-2" n1="18" n2="19" p1="50" p2="2"
 s1="1" s2="5" status="" .../>
   <fabricLink dn="topology/lnkcnt-20/lnk-18-1-49-to-20-5-1" n1="18" n2="20" p1="49" p2="1"
 s1="1" s2="5" status="" .../>
    <fabricLink dn="topology/lnkcnt-3/lnk-18-1-1-to-3-1-1" n1="18" n2="3" p1="1" p2="1"
s1="1" s2="1" status="" .../>
   <fabricLink dn="topology/lnkcnt-19/lnk-17-1-49-to-19-5-1" n1="17" n2="19" p1="49" p2="1"
 s1="1" s2="5" status="" .../>
   <fabricLink dn="topology/lnkcnt-20/lnk-17-1-50-to-20-5-2" n1="17" n2="20" p1="50" p2="2"
 s1="1" s2="5" status="" .../>
    <fabricLink dn="topology/lnkcnt-1/lnk-17-1-1-to-1-1-1" n1="17" n2="1" p1="1" p2="1"
s1="1" s2="1" status="" .../>
    <fabricLink dn="topology/lnkcnt-2/lnk-17-1-2-to-2-1-1" n1="17" n2="2" p1="2" p2="1"
s1="1" s2="1" status="" .../>
</imdata>
```

Unmanaged Switch Connectivity in VM Management

The hosts that are managed by the VM controller (for example, a vCenter), can be connected to the leaf port through a Layer 2 switch. The only prerequisite required is that the Layer 2 switch must be configured with a management address, and this management address must be advertised by Link Layer Discovery Protocol (LLDP) or Cisco Discovery Protocol (CDP) on the ports that are connected to the switches. Layer 2 switches are automatically discovered by the APIC, and they are identified by the management address. The following figure shows the APIC GUI displaying unmanaged switches in the **Fabric** > **Inventory** view.

Figure 1: Unmanaged Layer 2 Switches in the APIC Fabric Inventory



Unmanaged Fabric Node - 172.23.53.176 (ifav101-n3k-1)

PROPERTIES

System Name: **ifav101-n3k-1**

Management IP: **172.23.53.176**

System Description: **Cisco Nexus Operating System (NX-OS) Software 6.0(2)U4(2) TAC support: http://www.cisco.com/tac Copyright (c) 2002-2014, Cisco Systems, Inc. All rights reserved.**

Fabric Node Link:

NODE ID	NODE NAME	INTERFACE
101	ifav101-leaf1	eth1/45
101	ifav101-leaf1	eth1/46
101	ifav101-leaf1	eth1/48
102	ifav101-leaf2	eth1/30

Server Link:

SERVER	PNIC

No items have been found.
Select Actions to create a new item.

Configuring Network Time Protocol

Time Synchronization and NTP

Within the Cisco Application Centric Infrastructure (ACI) fabric, time synchronization is a crucial capability upon which many of the monitoring, operational, and troubleshooting tasks depend. Clock synchronization is important for proper analysis of traffic flows as well as for correlating debug and fault time stamps across multiple fabric nodes.

An offset present on one or more devices can hamper the ability to properly diagnose and resolve many common operational issues. In addition, clock synchronization allows for the full utilization of the atomic counter capability that is built into the ACI upon which the application health scores depend. Nonexistent or improper configuration of time synchronization does not necessarily trigger a fault or a low health score. You should configure time synchronization before deploying a full fabric or applications so as to enable proper usage of these features. The most widely adapted method for synchronizing a device clock is to use Network Time Protocol (NTP).

Prior to configuring NTP, consider what management IP address scheme is in place within the ACI fabric. There are two options for configuring management of all ACI nodes and Application Policy Infrastructure Controllers (APICs), in-band management and/or out-of-band management. Depending upon which management option is chosen for the fabric, configuration of NTP will vary. Another consideration in deploying time synchronization is where the time source is located. The reliability of the source must be carefully considered when determining if you will use a private internal clock or an external public clock.

In-Band and Out-of-Band Management NTP

Note

- Make sure the Management EPG is configured for the NTP servers, otherwise the servers will not get configured on the switches.
- See the Adding Management Access section in this guide for information about in-band management access and out-of-band management access.

- Out-of-band management NTP—When an ACI fabric is deployed with out-of-band management, each node of the fabric, inclusive of spines, leaves, and all members of the APIC cluster, is managed from outside the ACI fabric. This IP reachability will be leveraged so that each node can individually query the same NTP server as a consistent clock source. To configure NTP, a Date and Time policy must be created that references an out-of-band management endpoint group. Date and Time policies are confined to a single pod and must be deployed across all pods provisioned in the ACI fabric. Currently only one pod per ACI fabric is allowed.
- In-Band Management NTP—When an ACI fabric is deployed with in-band management, consider the reachability of the NTP server from within the ACI in-band management network. In-band IP addressing used within the ACI fabric is not reachable from anywhere outside the fabric. To leverage an NTP server external to the fabric with in-band management, construct a policy to enable this communication. The steps used to configure in-band management policies are identical to those used to establish an out-of-band management policy. The distinction is around how to allow the fabric to connect to the NTP server.

NTP over IPv6

NTP over IPv6 addresses is supported in hostnames and peer addresses. The gai.conf can also be set up to prefer the IPv6 address of a provider or a peer over an IPv4 address. The user can provide a hostname that can be resolved by providing an IP address (both IPv4 or IPv6, depending on the installation or preference).

Configuring NTP Using the REST API

Procedure

Step 1 Configure NTP.

Example:

```
POST url: https://APIC-IP/api/node/mo/uni/fabric/time-test.xml

<imdata totalCount="1">
   <datetimePol adminSt="enabled" authSt="disabled" descr="" dn="uni/fabric/time-CiscoNTPPol"
 name="CiscoNTPPol" ownerKey="" ownerTag="">
        <datetimeNtpProv descr="" keyId="0" maxPoll="6" minPoll="4" name="10.10.10.11"
preferred="yes">
            <datetimeRsNtpProvToEpg tDn="uni/tn-mgmt/mgmtp-default/inb-default"/>
        </datetimeNtpProv>
```

```
    </datetimePol>
</imdata>
```

Step 2 Add the default Date Time Policy to the pod policy group.

Example:

```
POST url: https://APIC-IP/api/node/mo/uni/fabric/funcprof/podpgrp-calo1/rsTimePol.xml

POST payload: <imdata totalCount="1">
<fabricRsTimePol tnDatetimePolName="CiscoNTPPol">
</fabricRsTimePol>
</imdata>
```

Step 3 Add the pod policy group to the default pod profile.

Example:

```
POST url:
https://APIC-IP/api/node/mo/uni/fabric/podprof-default/pods-default-typ-ALL/rspodPGrp.xml

payload: <imdata totalCount="1">
<fabricRsPodPGrp tDn="uni/fabric/funcprof/podpgrp-calo1" status="created">
</fabricRsPodPGrp>
</imdata>
```

Verifying NTP Operation Using the GUI

Procedure

Step 1 On the menu bar, choose **FABRIC** > **Fabric Policies**.

Step 2 In the **Navigation** pane, choose **Pod Policies** > **Policies** > **Date and Time** > ***ntp_policy*** > ***server_name***. The *ntp_policy* is the previously created policy. An IPv6 address is supported in the Host Name/IP address field. If you enter a hostname and it has an IPv6 address set, you must implement the priority of IPv6 address over IPv4 address.

Step 3 In the **Work** pane, verify the details of the server.

Verifying NTP Policy Deployed to Each Node Using the CLI

Procedure

Step 1 SSH to an APIC in the fabric.

Step 2 Press the Tab key two times after entering the attach command to list all the available node names:

Example:

```
admin@apic1:~> attach <Tab> <Tab>
```

Step 3 Log in one of the nodes using the same password that you used to access the APIC.

Example:

```
admin@apic1:~> attach node_name
```

Step 4 View the NTP peer status.

Example:

```
leaf-1# show ntp peer-status
```

A reachable NTP server has its IP address prefixed by an asterisk (*), and the delay is a non-zero value.

Step 5 Repeat steps 3 and 4 to verify each node in the fabric.

Creating User Accounts

Configuring a Local User

In the initial configuration script, the admin account is configured and the admin is the only user when the system starts. The APIC supports a granular, role-based access control system where user accounts can be created with various roles including non-admin users with fewer privileges.

Configuring a Remote User

Instead of configuring local users, you can point the APIC at the centralized enterprise credential datacenter. The APIC supports Lightweight Directory Access Protocol (LDAP), active directory, RADIUS, and TACACS+.

Note When an APIC is in minority (disconnected from the cluster), remote logins can fail because the ACI is a distributed system and the user information is distributed across APICS. Local logins, however, continue to work because they are local to the APIC.

To configure a remote user authenticated through an external authentication provider, you must meet the following prerequisites:

- The DNS configuration should have already been resolved with the hostname of the RADIUS server.
- You must configure the management subnet.

Configuring a Local User Using the REST API

Procedure

Create a local user.

Example:

```
URL: https://apic-ip-address/api/policymgr/mo/uni/userext.xml
POST CONTENT:
        <aaaUser name="operations" phone="" pwd="<strong_password>" >
           <aaaUserDomain childAction="" descr="" name="all" rn="userdomain-all" status="">

                 <aaaUserRole childAction="" descr="" name="Ops" privType="writePriv"/>
             </aaaUserDomain>
      </aaaUser>
```

AV Pair on the External Authentication Server

The Cisco APIC requires that an administrator configure a Cisco AV Pair on an external authentication server. The Cisco AV pair specifies the APIC required RBAC roles and privileges for the user. The Cisco AV Pair format is the same for RADIUS, LDAP, or TACACS+.

To configure a Cisco AV Pair on an external authentication server, an administrator adds a Cisco AV pair to the existing user record. The Cisco AV pair format is as follows:

```
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,
domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,
domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2(16003)
```

The first av-pair format has no UNIX user ID, while the second one does. Both are correct if all remote users have the same role and mutual file access is acceptable. If the UNIX user ID is not specified, ID 23999 is applied by the APIC system, and more than one role/read privilege is specified to any AV Pair user. This can cause users to have higher or lower permissions than configured through the group settings.

Note The APIC Cisco AV-pair format is compatible and can co-exist with other Cisco AV-pair formats. APIC will pick up the first matching AV-pair from all the AV-pairs.

The APIC supports the following regexes:

```
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})$
```

Examples:

- Example 1: A Cisco AV Pair that contains a single Login domain with only writeRoles:

  ```
  shell:domains=domainA/writeRole1|writeRole2/
  ```

- Example 2: A Cisco AV Pair that contains a single Login domain with only readRoles:

  ```
  shell:domains=domainA//readRole1|readRole2
  ```

Note The "/" character is a separator between writeRoles and readRoles per Login domain and is required even if only one type of role is to be used.

The Cisco AVpair string is case sensitive. Although a fault may not be seen, using mismatching cases for the domain name or roles could lead to unexpected privileges being given.

An example configuration for an open RADIUS server (/etc/raddb/users) is as follows:

```
aaa-network-admin Cleartext-Password := "<password>"
Cisco-avpair = "shell:domains = all/aaa/read-all(16001)"
```

Changing the Default Behavior for Remote Users with Missing or Bad Cisco AV Pairs

Procedure

Step 1 On the menu bar, click **ADMIN** > **AAA**.

Step 2 In the **Navigation** pane, click **AAA Authentication**.

Step 3 In the **Work** pane, in the **Properties** area, from the **Remote user login policy** drop-down list, choose **Assign Default Role**.
The default value is **No Login**. The **Assign Default Role** option assigns the minimal read-only privileges to users that have missing or bad Cisco AV Pairs. Bad AV Pairs are those AV Pairs that fail the parsing rules.

Best Practice for Assigning AV Pairs

As best practice, Cisco recommends that you assign unique UNIX user ids in the range 16000-23999 for the AV Pairs that are assigned to users when in bash shell (using SSH, Telnet or Serial/KVM consoles). If a situation arises when the Cisco AV Pair does not provide a UNIX user id, the user is assigned a user id of 23999 or similar number from the range that also enables the user's home directories, files, and processes accessible to remote users with a UNIX ID of 23999.

The Cisco AVpair string is case sensitive. Although a fault may not be seen, using mismatching cases for the domain name or roles could lead to unexpected privileges being given.

Configuring an AV Pair on the External Authentication Server

The numerical value within the parentheses in the attribute/value (AV) pair string is used as the UNIX user ID of the user who is logged in using Secure Shell (SSH) or Telnet.

Procedure

Configure an AV pair on the external authentication server.
The Cisco AV pair definition is as follows (Cisco supports AV pairs with and without UNIX user IDs specified):

Example:

```
* shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2

         * shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2(8101)

These are the boost regexes supported by APIC:
uid_regex("shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$");
regex("shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})$");
```

The following is an example:

```
shell:domains = coke/tenant-admin/read-all,pepsi//read-all(16001)
```

Configuring a Remote User Using the REST API

Procedure

Step 1 Create a RADIUS provider.

Example:

```
 URL: https://apic-ip-address/api/policymgr/mo/uni/userext/radiusext.xml
POST Content:
<aaaRadiusProvider name="radius-auth-server.org.com" key="test123" />
```

Step 2 Create a login domain.

Example:

```
 URL: https://apic-ip-address/api/policymgr/mo/uni/userext.xml
POST Content:
<aaaLoginDomain name="rad"> <aaaDomainAuth realm="radius"/> </aaaLoginDomain>
```

Adding Management Access

An APIC controller has two routes to reach the management network, one is by using the in-band management interface and the other is by using the out-of-band management interface.

- In-band management access—You can configure in-band management connectivity to the APIC and the ACI fabric. You first configure the VLANs that will be used by APIC when the APIC is communicating with the leaf switches, and then you configure the VLANs that the VMM servers will use to communicate with the leaf switches.
- Out-of-band management access—You can configure out-of-band management connectivity to the APIC and the ACI fabric. You configure an out-of-band contract that is associated with an out-of-band endpoint group (EPG), and attach the contract to the external network profile.

Note The APIC out-of-band management connection link must be 1 Gbps.

The APIC controller always selects the in-band management interface over the out-of-band management interface, if the in-band management interface is configured. The out-of-band management interface is used only when the in-band management interface is not configured, or if the destination address is on the same subnet as the out-of-band management subnet of the APIC.

The APIC management interface does not support an IPv6 address and cannot connect to an external IPv6 server through this interface.

Configuring the external management instance profile under the management tenant for in-band or out-of-band has no effect on the protocols that are configured under the fabric-wide communication policies. The subnets and contracts specified under the external management instance profile do not affect HTTP/HTTPS or SSH/Telnet.

IPv4/IPv6 Addresses and In-Band Policies

In-band management addresses can be provisioned on the APIC controller only through a policy (Postman REST API, NX-OS Style CLI, or GUI). Additionally, the in-band management addresses must be configured statically on each node.

IPv4/IPv6 Addresses in Out-of-Band Policies

Out-of-band management addresses can be provisioned on the APIC controller either at the time of bootstrap or by using a policy (Postman REST API, NX-OS Style CLI, GUI). Additionally, the out-of-band management addresses must be configured statically on each node or by specifying a range of addresses (IPv4/IPv6) to the entire cluster. IP addresses are randomly assigned from a range to the nodes in the cluster.

Configuring Management Access

Configuring In-Band Management Access Using the REST API

IPv4 and IPv6 addresses are supported for in-band management access. IPv6 configurations are supported using static configurations (for both in-band and out-of-band). IPv4 and IPv6 dual in-band and out-of-band configurations are supported only through static configuration. For more information, see the KB article,*Configuring Static Management Access in Cisco APIC*.

Procedure

Step 1 Create a VLAN namespace.

Example:

```
POST
https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/uni.xml -->
<polUni>
  <infraInfra>
    <!-- Static VLAN range -->
    <fvnsVlanInstP name="inband" allocMode="static">
      <fvnsEncapBlk name="encap" from="vlan-10" to="vlan-11"/>
    </fvnsVlanInstP>
  </infraInfra>
</polUni>
```

Step 2 Create a physical domain.

Example:

```
POST
https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/uni.xml -->
<polUni>
  <physDomP name="inband">
    <infraRsVlanNs tDn="uni/infra/vlanns-inband-static"/>
  </physDomP>
</polUni>
```

Step 3 Create selectors for the in-band management.

Example:

```
POST
https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <infraInfra>
    <infraNodeP name="vmmNodes">
      <infraLeafS name="leafS" type="range">
        <infraNodeBlk name="single0" from_="101" to_="101"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-vmmPorts"/>
    </infraNodeP>

    <!-- Assumption is that VMM host is reachable via eth1/40. -->
    <infraAccPortP name="vmmPorts">
      <infraHPortS name="portS" type="range">
        <infraPortBlk name="block1"
                      fromCard="1" toCard="1"
                      fromPort="40" toPort="40"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-inband" />
      </infraHPortS>
    </infraAccPortP>


    <infraNodeP name="apicConnectedNodes">
      <infraLeafS name="leafS" type="range">
        <infraNodeBlk name="single0" from_="101" to_="102"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-apicConnectedPorts"/>
    </infraNodeP>

    <!-- Assumption is that APIC is connected to eth1/1. -->
    <infraAccPortP name="apicConnectedPorts">
      <infraHPortS name="portS" type="range">
        <infraPortBlk name="block1"
                      fromCard="1" toCard="1"
                      fromPort="1" toPort="3"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-inband" />
      </infraHPortS>
    </infraAccPortP>

    <infraFuncP>
      <infraAccPortGrp name="inband">
        <infraRsAttEntP tDn="uni/infra/attentp-inband"/>
      </infraAccPortGrp>
    </infraFuncP>

    <infraAttEntityP name="inband">
```

```
      <infraRsDomP tDn="uni/phys-inband"/>
    </infraAttEntityP>
  </infraInfra>
</polUni>
```

Step 4 Configure an in-band bridge domain and endpoint group (EPG).

Example:

```
POST https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="mgmt">
    <!-- Configure the in-band management gateway address on the
         in-band BD. -->
    <fvBD name="inb">
      <fvSubnet ip="10.13.1.254/24"/>
    </fvBD>

    <mgmtMgmtP name="default">
      <!-- Configure the encap on which APICs will communicate on the
           in-band network. -->
      <mgmtInB name="default" encap="vlan-10">
        <fvRsProv tnVzBrCPName="default"/>
      </mgmtInB>
    </mgmtMgmtP>
  </fvTenant>
</polUni>
```

Step 5 Create an address pool.

Example:

```
POST
https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="mgmt">
    <!-- Adresses for APIC in-band management network -->
    <fvnsAddrInst name="apicInb" addr="10.13.1.254/24">
      <fvnsUcastAddrBlk from="10.13.1.1" to="10.13.1.10"/>
    </fvnsAddrInst>

    <!-- Adresses for switch in-band management network -->
    <fvnsAddrInst name="switchInb" addr="10.13.1.254/24">
      <fvnsUcastAddrBlk from="10.13.1.101" to="10.13.1.120"/>
    </fvnsAddrInst>
  </fvTenant>
</polUni>
```

Note Dynamic address pools for IPv6 is not supported.

Step 6 Create management groups.

Example:

```
POST
https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <infraInfra>
    <!-- Management node group for APICs -->
    <mgmtNodeGrp name="apic">
      <infraNodeBlk name="all" from_="1" to_="3"/>
      <mgmtRsGrp tDn="uni/infra/funcprof/grp-apic"/>
    </mgmtNodeGrp>
    <!-- Management node group for switches-->
    <mgmtNodeGrp name="switch">
      <infraNodeBlk name="all" from_="101" to_="104"/>
      <mgmtRsGrp tDn="uni/infra/funcprof/grp-switch"/>
    </mgmtNodeGrp>

    <!-- Functional profile -->
    <infraFuncP>
      <!-- Management group for APICs -->
      <mgmtGrp name="apic">
        <!-- In-band management zone -->
        <mgmtInBZone name="default">
          <mgmtRsInbEpg tDn="uni/tn-mgmt/mgmtp-default/inb-default"/>
          <mgmtRsAddrInst tDn="uni/tn-mgmt/addrinst-apicInb"/>
        </mgmtInBZone>
      </mgmtGrp>

      <!-- Management group for switches -->
      <mgmtGrp name="switch">
        <!-- In-band management zone -->
        <mgmtInBZone name="default">
          <mgmtRsInbEpg tDn="uni/tn-mgmt/mgmtp-default/inb-default"/>
          <mgmtRsAddrInst tDn="uni/tn-mgmt/addrinst-switchInb"/>
        </mgmtInBZone>
      </mgmtGrp>
    </infraFuncP>
  </infraInfra>
</polUni>
```

Note Dynamic address pools for IPv6 is not supported.

Configuring Out-of-Band Management Access Using the REST API

IPv4 and IPv6 addresses are supported for out-of-band management access.

Before You Begin

The APIC out-of-band management connection link must be 1 Gbps.

Procedure

Step 1 Create an out-of-band contract.

Example:

```
POST https://apic-ip-address/api/mo/uni.xml

<polUni>
    <fvTenant name="mgmt">
        <!-- Contract -->
        <vzOOBBrCP name="oob-default">
            <vzSubj name="oob-default">
                <vzRsSubjFiltAtt tnVzFilterName="default" />
            </vzSubj>
        </vzOOBBrCP>
    </fvTenant>
</polUni>
```

Step 2 Associate the out-of-band contract with an out-of-band EPG.

Example:

```
POST https://apic-ip-address/api/mo/uni.xml

<polUni>
    <fvTenant name="mgmt">
        <mgmtMgmtP name="default">
            <mgmtOoB name="default">
                <mgmtRsOoBProv tnVzOOBBrCPName="oob-default" />
            </mgmtOoB>
        </mgmtMgmtP>
    </fvTenant>
</polUni>
```

Step 3 Associate the out-of-band contract with an external management EPG.

Example:

```
POST https://apic-ip-address/api/mo/uni.xml

<polUni>
    <fvTenant name="mgmt">
        <mgmtExtMgmtEntity name="default">
            <mgmtInstP name="oob-mgmt-ext">
                <mgmtRsOoBCons tnVzOOBBrCPName="oob-default" />
                <!-- SUBNET from where switches are managed -->
                <mgmtSubnet ip="10.0.0.0/8" />
            </mgmtInstP>
        </mgmtExtMgmtEntity>
    </fvTenant>
</polUni>
```

Step 4 Create a management address pool.

Example:

```
POST https://apic-ip-address/api/mo/uni.xml

<polUni>
    <fvTenant name="mgmt">
        <fvnsAddrInst name="switchOoboobaddr" addr="172.23.48.1/21">
            <fvnsUcastAddrBlk from="172.23.49.240" to="172.23.49.244"/>
        </fvnsAddrInst>
    </fvTenant>
</polUni>
```

Step 5 Create node management groups.

Example:

```
POST https://apic-ip-address/api/mo/uni.xml

<polUni>
    <infraInfra>
        <infraFuncP>
            <mgmtGrp name="switchOob">
                <mgmtOoBZone name="default">
                    <mgmtRsAddrInst tDn="uni/tn-mgmt/addrinst-switchOoboobaddr" />
                    <mgmtRsOobEpg tDn="uni/tn-mgmt/mgmtp-default/oob-default" />
                </mgmtOoBZone>
            </mgmtGrp>
        </infraFuncP>
        <mgmtNodeGrp name="switchOob">
            <mgmtRsGrp tDn="uni/infra/funcprof/grp-switchOob" />
            <infraNodeBlk name="default" from_="101" to_="103" />
        </mgmtNodeGrp>
    </infraInfra>
```

```
</polUni>
```

Note You can configure the APIC server to use out-of-band management connectivity as the default connectivity mode.

```
POST https://apic-ip-address/api/node/mo/.xml
<polUni>
<fabricInst>
    <mgmtConnectivityPrefs interfacePref=“ooband"/>
</fabricInst>
</polUni>
```

Modifying the IP Address of an APIC Controller Using the REST API

Procedure

Modify the IP address of the APIC controller.

Example:

```
POST
https://APIC-IP/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- /api/policymgr/mo/.xml -->
<polUni>
  <infraInfra>
    <infraFuncP>
      <mgmtGrp name="mgmtGroupApic">
        <mgmtOoBZone name="mgmtOobZoneApic">
          <mgmtRsOobEpg tDn="uni/tn-mgmt/mgmtp-default/oob-default"/>
          <mgmtRsAddrInst tDn="uni/tn-mgmt/addrinst-oobAddrApic"/>
        </mgmtOoBZone>
      </mgmtGrp>
    </infraFuncP>
    <mgmtNodeGrp name="mgmtNodeGroupApic">
      <mgmtRsGrp tDn="uni/infra/funcprof/grp-mgmtGroupApic"/>
      <infraNodeBlk name="default" from_="1" to_="1"/>
    </mgmtNodeGrp>
  </infraInfra>
</polUni>

<?xml version="1.0" encoding="UTF-8"?>
<!-- /api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="mgmt">
    <fvnsAddrInst name="oobAddrApic" addr="172.23.48.1/21">
      <fvnsUcastAddrBlk from="172.23.48.16" to="172.23.48.16"/>
    </fvnsAddrInst>
  </fvTenant>
</polUni>
```

What to Do Next

- You must use the new IP address to reconnect to the APIC controller.
- You must delete the old IP address of the controller once a new IP address is assigned to it.

IPv6 Table Modifications to Mirror the Existing IP Tables Functionality

All IPv6 tables mirror the existing IP tables functionality, except for Network Address Translation (NAT).

Existing IP Tables

1. Earlier, every rule in the IPv6 tables were executed one at a time and a system call was made for every rule addition or deletion.
2. Whenever a new policy was added, rules were appended to the existing IP tables file and no extra modifications were done to the file.
3. When a new source port was configured in the out-of-band policy, it added source and destination rules with the same port number.

Modifications to IP Tables

1. When IP tables are created, they are first written into hash maps that are then written into intermediate file IP tables-new which are restored. When saved, a new IP tables file is created in the /etc/sysconfig/ folder. You can find both these files at the same location. Instead of making a system call for every rule, you must make a system call only while restoring and saving the file.
2. When a new policy is added instead of appending it to the file, an IP table is created from scratch, that is by loading default policies into the hashmaps, checking for new policies, and adding them to hashmaps. Later, they are written to the intermediate file (/etc/sysconfig/iptables-new) and saved.
3. It is not possible to configure source ports alone for a rule in out-of-band policy. Either destination port or source port along with a destination port can be added to the rules.
4. When a new policy is added, a new rule will be added to the IP tables file. This rule changes the access flow of IP tables default rules.

   ```
   -A INPUT -s <OOB Address Ipv4/Ipv6> -j apic-default
   ```
5. When a new rule is added, it presents in the IP tables-new file and not in the IP tables file, and it signifies that there is some error in the IP tables-new file. Only if the restoration is successful, the file is saved and new rules are seen in the IP tables file.

Note

- If only IPv4 is enabled, do not configure an IPv6 policy.
- If only IPv6 is enabled, do not configure an IPv4 policy.
- If both IPv4 and IPv6 are enabled and a policy is added, it will be configured to both the versions . So when you add an IPv4 subnet, it will be added to IP tables and similarly an IPv6 subnet is added to IPv6 tables.

Management Connectivity Modes

Establish connection to external entities using the out-of-band or in-band network depending upon whether you have configured out-of-band and/or in-band management connectivity. The following two modes are available to establish connectivity to external entities such as the vCenter server:

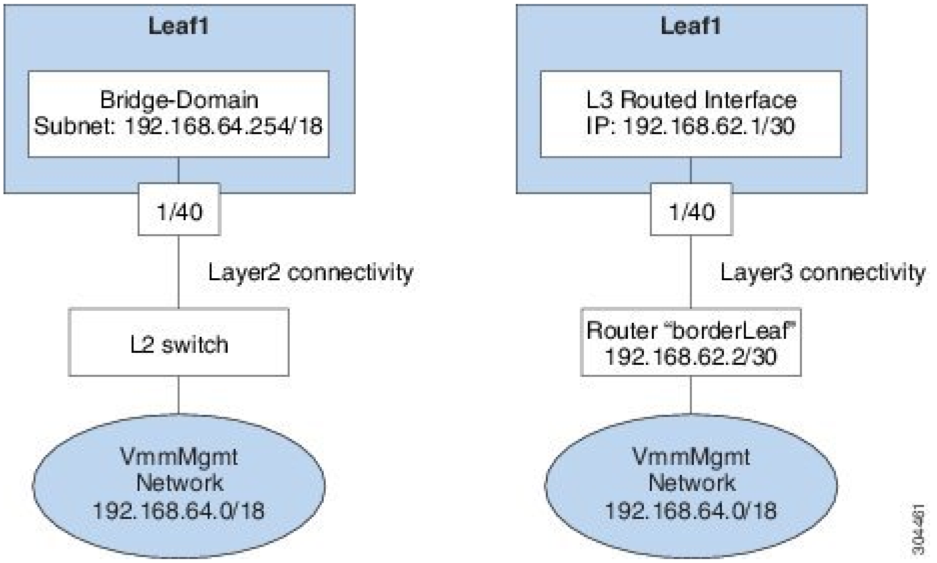
- Layer 2 management connectivity—Use this mode when the external entities are attached to the leaf node using Layer 2.
- Layer 3 management connectivity—Use this mode when the external entities are attached to the leaf node using Layer 3 through a router. The leaf is connected to a router through which external entities can be reached.

Note

- The inband IP address range must be separate and distinct from the IP address range used on the Layer 3 connection from the leaf node to outside the fabric.
- The Layer 3 inband management design does not provide inband management access to the spine fabric nodes in the topology.

The following diagram displays the two modes available to establish connectivity.

Figure 2: Layer 2 and Layer 3 Management Connectivity Examples



Configuring Layer 2 Management Connectivity Using the REST API

Note The name vmm is used as an example string in this task.

The policy creates the following objects under Tenant-mgmt:

Creates bridge domain (vmm) and the following related objects as follows:

- Creates the subnet object with this IP prefix (192.168.64.254/18) in this bridge domain. This IP address (192.168.64.254) is assigned to the bridge domain that typically is used as the switch virtual interface (SVI) in a traditional switch configuration.
- Creates an association to the in-band network (ctx).

Creates an application profile (vmm) and management EPG (vmmMgmt) with related objects as follows:

- Creates an association to the bridge domain (vmm).
- Creates a policy to deploy this EPG on leaf1. The encapsulation used for this EPG is vlan-11.

Before You Begin

Before you create a vCenter domain profile, you must establish connectivity to establish an external network using an in-band management network.

Make sure that the IP address range configured as part of management connectivity policy does not overlap with the infrastructure IP address range used by the ACI fabric.

Procedure

You can establish connectivity from the APIC to external routes by using a router that is connected to a leaf port.

Example:

```
POST
https://APIC-IP/api/mo/uni.xml

<!-- api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="mgmt">
    <fvBD name="vmm">
      <fvRsCtx tnFvCtxName="inb"/>
      <fvSubnet ip='192.168.64.254/18'/>
    </fvBD>

    <fvAp name="vmm">
      <fvAEPg name="vmmMgmt">
        <fvRsBd tnFvBDName="vmm" />
        <fvRsPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/40]" encap="vlan-11"/>
        <fvRsCons tnVzBrCPName="default"/>
        <fvRsDomAtt tDn="uni/phys-inband"/>
      </fvAEPg>
    </fvAp>
  </fvTenant>
</polUni>
```

Configuring Layer 3 Management Connectivity Using the REST API

The name vmm is used as an example string in this task.

The policy creates the following objects under Tenant-mgmt:

- Creates a routed outside policy (vmm) with the following details:

  1 Creates a Layer 3 external network instance profile object (vmmMgmt).

2 Creates a route for the remote network (192.168.64.0/18) with the IP address of the next-hop router 192.168.62.2.

3 Creates a logical node profile object (borderLeaf) that is attached to leaf1.

4 Creates a port profile (portProfile1) with the routed interface 1/40 with the IP address 192.168.62.1/30.

5 Creates an association to inband network (ctx).

Before You Begin

Make sure that the IP address range configured as part of management connectivity policy does not overlap with the infrastructure IP address range used by the ACI fabric.

Procedure

You can establish connectivity from the APIC to external routes by using a router that is connected to a leaf port.

Example:

```
<!-- api/policymgr/mo/.xml -->
POST
https://APIC-IP/api/mo/uni.xml

<polUni>
   <fvTenant name="mgmt">
      <l3extOut name="vmm">
         <l3extInstP name="vmmMgmt">
            <l3extSubnet ip="192.168.0.0/16" />
            <fvRsCons tnVzBrCPName="default" />
         </l3extInstP>
         <l3extLNodeP name="borderLeaf">
            <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="1.2.3.4">
               <ipRouteP ip="192.168.64.0/18">
                  <ipNexthopP nhAddr="192.168.62.2" />
               </ipRouteP>
            </l3extRsNodeL3OutAtt>
            <l3extLIfP name="portProfile">
               <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/40]"
ifInstT="l3-port" addr="192.168.62.1/30" />
            </l3extLIfP>
         </l3extLNodeP>
         <l3extRsEctx tnFvCtxName="inb" />
      </l3extOut>
   </fvTenant>
</polUni>
```

Validating Management Connectivity

This validation process applies to both Layer 2 and Layer 3 modes and can be used to verify connectivity that is established by using the APIC GUI, REST API, or CLI.

After completing the steps to establish management connectivity, log in to the APIC console. Ping to the IP address of the vCenter server that is reachable (for example, 192.168.81.2) and verify that the ping works. This action indicates that the policies have been successfully applied.

Configuring a VMM Domain

Configuring Virtual Machine Networking Policies

The APIC integrates with third-party VM manager (VMM) (for example, VMware vCenter and SCVMM) to extend the benefits of ACI to the virtualized infrastructure. The APIC enables the ACI policies inside the VMM system to be used by its administrator.

This section provides examples of VMM integration using VMware vCenter and vShield. For details about the different modes of Cisco ACI and VMM integration, see the *ACI Virtualization Guide*.

About the VM Manager

Note Information about the necessary configuration of the APIC for integration with the vCenter is described here. For instructions about configuring the VMware components, see the VMware documentation.

The following are details of some VM manager terms:

- A VM controller is an external virtual machine management entity such as VMware vCenter, and the VMware vShield. The APIC communicates with the controller to publish network policies that are applied to virtual workloads. A VM controller administrator provides an APIC administrator with a VM controller authentication credential; multiple controllers of the same type can use the same credential.
- Credentials represent the authentication credentials to communicate with VM controllers. Multiple controllers can use the same credentials.
- A virtual machine mobility domain (vCenter mobility domain) is a grouping of VM controllers with similar networking policy requirements. This mandatory container holds one or more VM controllers with policies such as for a VLAN pool, server to network MTU policy, or server to network access LACP policy. When an endpoint group gets associated with a vCenter domain, network policies get pushed to all the VM controllers in the vCenter domain.
- A pool represents a range of traffic encapsulation identifiers (for example, VLAN IDs, VNIDs, and multicast addresses). A pool is a shared resource and can be consumed by multiple domains such as VMM and Layer 4 to Layer 7 services. A leaf switch does not support overlapping VLAN pools. You must not associate different overlapping VLAN pools with the VMM domain. The two types of VLAN-based pools are as follows:
    - Dynamic pools—Managed internally by the APIC to allocate VLANs for endpoint groups (EPGs). A vCenter Domain can associate only to a dynamic pool.
    - Static pools—The EPG has a relation to the domain, and the domain has a relation to the pool. The pool contains a range of encapsulated VLANs and VXLANs. For static EPG deployment, the user defines the interface and the encapsulation. The encapsulation must be within the range of a pool that is associated with a domain with which the EPG is associated.
- For a VMware vCenter to be deployed, it must operate in VLAN mode or VXLAN mode. A VMM domain must be associated with a VLAN pool and a vShield must be associated with the vCenter.

About Attachable Entity Profile

Attach Entity Profiles

The ACI fabric provides multiple **attachment points** that connect through leaf ports to various **external entities** such as baremetal servers, hypervisors, Layer 2 switches (for example, the Cisco UCS fabric interconnect), and Layer 3 routers (for example Cisco Nexus 7000 Series switches). These attachment points can be physical ports, port channels, or a virtual port channel (vPC) on the leaf switches.

An **attachable entity profile** (AEP) represents a group of external entities with similar infrastructure policy requirements. The infrastructure policies consist of physical interface policies, for example, Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol (LLDP), maximum transmission unit (MTU), and Link Aggregation Control Protocol (LACP).

A VM manager (VMM) domain automatically derives the physical interfaces policies from the interface policy groups that are associated with an AEP.

- An override policy at AEP can be used to specify a different physical interface policy for a VMM domain. This policy is useful in scenarios where a hypervisor is connected to the leaf switch through an intermediate Layer 2 node, and a different policy is desired at the leaf switch and hypervisor physical ports. For example, you can configure LACP between a leaf switch and a Layer 2 node. At the same time, you can disable LACP between the hypervisor and the Layer 2 switch by disabling LACP under the AEP override policy.

An AEP is required to deploy any VLAN pools on the leaf switches. It is possible to reuse the encapsulation pools (for example, VLAN) across different leaf switches. An AEP implicitly provides the scope of the VLAN pool (associated to the domain) to the physical infrastructure.

Note

- An AEP provisions the VLAN pool (and associated VLANs) on the leaf. The VLANs are not actually enabled on the port. No traffic flows unless an EPG is deployed on the port.
- Without VLAN pool deployment using an AEP, a VLAN is not enabled on the leaf port even if an EPG is provisioned.
    - A particular VLAN is provisioned or enabled on the leaf port based on EPG events either statically binding on a leaf port or based on VM events from external controllers such as VMware vCenter.
    - If you wish to set the VMM encapsulation statically in the EPG, you must use a static pool. If you have a mix of static and dynamic allocations, create a dynamic pool and add a block within that pool with static mode.
- A leaf switch does not support overlapping VLAN pools. Different overlapping VLAN pools must not be associated with the same AEP that is associated through a domain.

For information about configuring LLDP and CDP, see the chapter related to Working with Blade Servers in the guide.

Prerequisites for Creating a VMM Domain Profile

To configure a VMM domain profile, you must meet the following prerequisites:

- All fabric nodes are discovered and configured.
- Inband (inb) or out-of-band (oob) management has been configured on the APIC.
- A Virtual Machine Manager (VMM) is installed, configured, and reachable through the inb/oob management network (for example, a vCenter).
- You have the administrator/root credentials to the VMM (for example vCenter).

  Note If you prefer not to use the vCenter admin/root credentials, you can create a custom user account with minimum required permissions. See Custom User Account with Minimum VMware vCenter Privileges for a list of the required user privileges.

- A DNS policy for the APIC must be configured if you plan to reference the VMM by hostname rather than an IP address.

Custom User Account with Minimum VMware vCenter Privileges

This allows the APIC to send VMware API commands to vCenter to allow the creation of the DVS/AVS, creation of the VMK interface (AVS), publish port groups and relay all necessary alerts.

To configure the vCenter from Cisco APIC, your credentials must allow the following minimum set of privileges within the vCenter:

- **Alarms**

  APIC creates two alarms on the folder. One for DVS and another for port-group. The alarm is raised when the EPG or Domain policy is deleted on APIC, but for port-group or DVS it cannot be deleted due to the VMs are attached.

- **Distributed Switch**
- **dvPort Group**
- **Folder**
- **Network**

  APIC manages the network settings such as add or delete port-groups, setting host/DVS MTU, LLDP/CDP, LACP etc.

- **Host**

  If you use AVS in addition to above, you need the Host privilege on the data center where APIC will create DVS.

  - **Host.Configuration.Advanced settings**
  - **Host.Local operations.Reconfigure virtual machine**
  - **Host.Configuration.Network configuration**

This is needed for AVS and the auto-placement feature for virtual Layer 4 to Layer 7 Service VMs. For AVS, APIC creates VMK interface and places it in ‘vtep’ port-group which is used for OpFlex.

- **Virtual machine**

  If you use Service Graph in addition to above, you need the Virtual machine privilege for the virtual appliances which will be used for Service Graph.

  - **Virtual machine.Configuration.Modify device settings**
  - **Virtual machine.Configuration.Settings**

Creating a VMM Domain Profile

In this section, examples of a VMM domain are vCenter domain.

Creating a vCenter Domain Profile Using the REST API

Procedure

Step 1 Configure a VMM domain name, a controller, and user credentials.

Example:

```
 POST URL: https://<api-ip>/api/node/mo/.xml

<polUni>
<vmmProvP vendor="VMware">
<!-- VMM Domain -->
<vmmDomP name="productionDC">
<!-- Association to VLAN Namespace -->
<infraRsVlanNs tDn="uni/infra/vlanns-VlanRange-dynamic"/>
<!-- Credentials for vCenter -->
<vmmUsrAccP name="admin" usr="administrator" pwd="admin" />
<!-- vCenter IP address -->
<vmmCtrlrP name="vcenter1" hostOrIp="<vcenter ip address>" rootContName="<Datacenter Name
in vCenter>">
<vmmRsAcc tDn="uni/vmmp-VMware/dom-productionDC/usracc-admin"/>
</vmmCtrlrP>
</vmmDomP>
</vmmProvP>
```

Example:

```
<polUni>
<vmmProvP vendor="VMware">
    <vmmDomP name=“mininet" delimiter=“@" >
   </vmmDomP>
</vmmProvP>
</polUni>
```

Step 2 Create an attachable entity profile for VLAN namespace deployment.

Example:

```
POST URL: https://<apic-ip>/api/policymgr/mo/uni.xml
<infraInfra>
```

```
<infraAttEntityP name="profile1">
<infraRsDomP tDn="uni/vmmp-VMware/dom-productionDC"/>
</infraAttEntityP>
</infraInfra>
```

Step 3 Create an interface policy group and selector.

Example:

```
POST URL: https://<apic-ip>/api/policymgr/mo/uni.xml

<infraInfra>
    <infraAccPortP name="swprofile1ifselector">
        <infraHPortS name="selector1" type="range">
            <infraPortBlk name="blk"
             fromCard="1" toCard="1" fromPort="1" toPort="3">
            </infraPortBlk>
     <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-group1" />
        </infraHPortS>
    </infraAccPortP>

    <infraFuncP>
        <infraAccPortGrp name="group1">
            <infraRsAttEntP tDn="uni/infra/attentp-profile1" />
        </infraAccPortGrp>
    </infraFuncP>
</infraInfra>
```

Step 4 Create a switch profile.

Example:

```
POST URL: https://<apic-ip>/api/policymgr/mo/uni.xml

<infraInfra>
    <infraNodeP name="swprofile1">
        <infraLeafS name="selectorswprofile11718" type="range">
            <infraNodeBlk name="single0" from_="101" to_="101"/>
            <infraNodeBlk name="single1" from_="102" to_="102"/>
        </infraLeafS>
        <infraRsAccPortP tDn="uni/infra/accportprof-swprofile1ifselector"/>
    </infraNodeP>
</infraInfra>
```

Step 5 Configure the VLAN pool.

Example:

```
 POST URL: https://<apic-ip>/api/node/mo/.xml

<polUni>
<infraInfra>
<fvnsVlanInstP name="VlanRange" allocMode="dynamic">
   <fvnsEncapBlk name="encap" from="vlan-100" to="vlan-400"/>
</fvnsVlanInstP>
</infraInfra>
</polUni>
```

Step 6 Locate all the configured controllers and their operational state.

Example:

```
GET:
https://<apic-ip>/api/node/class/compCtrlr.xml?
<imdata>
<compCtrlr apiVer="5.1" ctrlrPKey="uni/vmmp-VMware/dom-productionDC/ctrlr-vcenter1"
deployIssues="" descr="" dn="comp/prov-VMware/ctrlr-productionDC-vcenter1" domName="
productionDC"
hostOrIp="esx1" mode="default" model="VMware vCenter Server 5.1.0 build-756313"
```

```
name="vcenter1" operSt="online" port="0" pwd="" remoteOperIssues="" scope="vm"
usr="administrator" vendor="VMware, Inc." ... />
</imdata>
```

Step 7 Locate the hypervisor and VMs for a vCenter with the name 'vcenter1' under a VMM domain called 'ProductionDC'.

Example:

```
GET:
https://<apic-ip>/api/node/mo/comp/prov-VMware/ctrlr-productionDC-vcenter1.xml?query-target=children

<imdata>
<compHv descr="" dn="comp/prov-VMware/ctrlr-productionDC-vcenter1/hv-host-4832" name="esx1"
state="poweredOn" type="hv" ... />
<compVm descr="" dn="comp/prov-VMware/ctrlr-productionDC-vcenter1/vm-vm-5531" name="AppVM1"
state="poweredOff" type="virt" .../>
<hvsLNode dn="comp/prov-VMware/ctrlr-productionDC-vcenter1/sw-dvs-5646" lacpEnable="yes"
lacpMode="passive" ldpConfigOperation="both" ldpConfigProtocol="lldp" maxMtu="1500"
mode="default" name="apicVswitch" .../>
</imdata>
```

Creating a vCenter and a vShield Domain Profile Using the REST API

Procedure

Step 1 Create a VLAN pool.

Example:

```
POST URL: https://<apic-ip>/api/policymgr/mo/uni.xml

<polUni>
    <infraInfra>
        <fvnsVlanInstP name="vlan1" allocMode="dynamic">
            <fvnsEncapBlk name="encap" from="vlan-100" to="vlan-400"/>
        </fvnsVlanInstP>
    </infraInfra>
</polUni>
```

Step 2 Create a vCenter domain, and assign a VLAN pool.

Example:

```
POST URL: https://<apic-ip>/api/policymgr/mo/uni.xml
<vmmProvP dn="uni/vmmp-VMware">
<vmmDomP name="productionDC">
<infraRsVlanNs tDn="uni/infra/vlanns-vlan1-dynamic"/>
</vmmDomP>
</vmmProvP>
```

Step 3 Create an attachable entity profile for infrastructure VLAN deployment.

Example:

```
POST URL: https://<apic-ip>/api/policymgr/mo/uni.xml

<infraInfra>
    <infraAttEntityP name="profile1">
        <infraRsDomP tDn="uni/vmmp-VMware/dom-productionDC"/>
        <infraProvAcc name="provfunc"/>
    </infraAttEntityP>
</infraInfra>
```

Step 4 Create an interface policy group and selector.

Example:

```
POST URL: https://<apic-ip>/api/policymgr/mo/uni.xml

<infraInfra>
    <infraAccPortP name="swprofile1ifselector">
        <infraHPortS name="selector1" type="range">
            <infraPortBlk name="blk"
             fromCard="1" toCard="1" fromPort="1" toPort="3">
            </infraPortBlk>
     <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-group1" />
        </infraHPortS>
    </infraAccPortP>

    <infraFuncP>
        <infraAccPortGrp name="group1">
            <infraRsAttEntP tDn="uni/infra/attentp-profile1" />
        </infraAccPortGrp>
    </infraFuncP>
</infraInfra>
```

Step 5 Create a switch profile.

Example:

```
POST URL: https://<apic-ip>/api/policymgr/mo/uni.xml

<infraInfra>
    <infraNodeP name="swprofile1">
        <infraLeafS name="selectorswprofile11718" type="range">
            <infraNodeBlk name="single0" from_="101" to_="101"/>
            <infraNodeBlk name="single1" from_="102" to_="102"/>
        </infraLeafS>
        <infraRsAccPortP tDn="uni/infra/accportprof-swprofile1ifselector"/>
    </infraNodeP>
</infraInfra>
```

Step 6 Create credentials for controllers.

Example:

```
POST URL: https://<apic-ip>/api/policymgr/mo/uni.xml

<vmmProvP dn="uni/vmmp-VMware">
    <vmmDomP name="productionDC">
        <vmmUsrAccP name="vcenter_user" usr="administrator" pwd="default"/>
        <vmmUsrAccP name="vshield_user" usr="admin" pwd="default"/>
    </vmmDomP>
</vmmProvP>
```

Step 7 Create a vCenter controller

Example:

```
<vmmProvP dn="uni/vmmp-VMware">
    <vmmDomP name="productionDC">
        <vmmCtrlrP name="vcenter1" hostOrIp="172.23.50.85" rootContName="Datacenter1">
            <vmmRsAcc tDn="uni/vmmp-VMware/dom-productionDC/usracc-vcenter_user"/>
        </vmmCtrlrP>
    </vmmDomP>
</vmmProvP>
```

Step 8 Create a VXLAN pool and a multicast address range.

Example:

```
POST URL: https://<apic-ip>/api/policymgr/mo/uni.xml
```

```
<infraInfra>
    <fvnsVxlanInstP name="vxlan1">
        <fvnsEncapBlk name="encap" from="vxlan-6000" to="vxlan-6200"/>
    </fvnsVxlanInstP>
    <fvnsMcastAddrInstP name="multicast1">
        <fvnsMcastAddrBlk name="mcast" from="224.0.0.1" to="224.0.0.20"/>
    </fvnsMcastAddrInstP>
</infraInfra>
```

Step 9 Create a vShield controller.

Example:

```
POST URL: https://<apic-ip>/api/policymgr/mo/uni.xml

<vmmProvP dn="uni/vmmp-VMware">
    <vmmDomP name="productionDC">
        <vmmCtrlrP name="vshield1" hostOrIp="172.23.54.62" scope="iaas">
            <vmmRsAcc tDn="uni/vmmp-VMware/dom-productionDC/usracc-vshield_user"/>
            <vmmRsVmmCtrlrP tDn="uni/vmmp-VMware/dom-productionDC/ctrlr-vcenter1"/>
            <vmmRsVxlanNs tDn="uni/infra/vxlanns-vxlan1"/>
            <vmmRsMcastAddrNs tDn="uni/infra/maddrns-multicast1"/>
        </vmmCtrlrP>
    </vmmDomP>
</vmmProvP>
```

Creating Tenants, VRF, and Bridge Domains

Tenants Overview

- A tenant contains policies that enable qualified users domain-based access control. Qualified users can access privileges such as tenant administration and networking administration.
- A user requires read/write privileges for accessing and configuring policies in a domain. A tenant user can have specific privileges into one or more domains.
- In a multitenancy environment, a tenant provides group user access privileges so that resources are isolated from one another (such as for endpoint groups and networking). These privileges also enable different users to manage different tenants.

Tenant Creation

A tenant contains primary elements such as filters, contracts, bridge domains, and application profiles that you can create after you first create a tenant.

VRF and Bridge Domains

You can create and specify a VRF and a bridge domain for the tenant. The defined bridge domain element subnets reference a corresponding Layer 3 context.

For details about enabling IPv6 Neighbor Discovery see*IPv6 and Neighbor Discovery* in *Cisco APIC Layer 3 Networking Guide*.

Creating a Tenant, VRF, and Bridge Domain Using the REST API

Procedure

Step 1 Create a tenant.

Example:

```
 POST https://apic-ip-address/api/mo/uni.xml
<fvTenant name="ExampleCorp"/>
```

When the POST succeeds, you see the object that you created in the output.

Step 2 Create a VRF and bridge domain.

Note The Gateway Address can be an IPv4 or an IPv6 address. For more about details IPv6 gateway address, see the related KB article, *KB: Creating a Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery* .

Example:

```
 URL for POST: https://apic-ip-address/api/mo/uni/tn-ExampleCorp.xml

<fvTenant name="ExampleCorp">
   <fvCtx name="pvn1"/>
   <fvBD name="bd1">
      <fvRsCtx tnFvCtxName="pvn1"/>
      <fvSubnet ip="10.10.100.1/24"/>
   </fvBD>
</fvTenant>
```

Note If you have a public subnet when you configure the routed outside, you must associate the bridge domain with the outside configuration.

Configuring Server or Service Policies

Configuring a DHCP Relay Policy

A DHCP relay policy may be used when the DHCP client and server are in different subnets. If the client is on an ESX hypervisor with a deployed vShield Domain profile, then the use of a DHCP relay policy configuration is mandatory.

When a vShield controller deploys a Virtual Extensible Local Area Network (VXLAN), the hypervisor hosts create a kernel (vmkN, virtual tunnel end-point [VTEP]) interface. These interfaces need an IP address in the infrastructure tenant that uses DHCP. Therefore, you must configure a DHCP relay policy so that the APIC can act as the DHCP server and provide these IP addresses.

When an ACI fabric acts as a DHCP relay, it inserts the DHCP Option 82 (the DHCP Relay Agent Information Option) in DHCP requests that it proxies on behalf of clients. If a response (DHCP offer) comes back from a DHCP server without Option 82, it is silently dropped by the fabric. Therefore, when the ACI fabric acts as a DHCP relay, DHCP servers providing IP addresses to compute nodes attached to the ACI fabric must support Option 82.

Configuring a DHCP Server Policy for the APIC Infrastructure Using the REST API

- This task is a prerequisite for users who want to create a vShield Domain Profile.
- The port and the encapsulation used by the application Endpoint Group must belong to a physical or VM Manager (VMM) domain. If no such association with a domain is established, the APIC continues to deploy the EPG but raises a fault.
- Cisco APIC supports DHCP relay for both IPv4 and IPv6 tenant subnets. DHCP server addresses can be IPv4 or IPv6. DHCPv6 relay will occur only if IPv6 is enabled on the fabric interface and one or more DHCPv6 relay servers are configured.

Before You Begin

Make sure that Layer 2 or Layer 3 management connectivity is configured.

Procedure

Configure the APIC as the DHCP server policy for the infrastructure tenant.

Note This relay policy will be pushed to all the leaf ports that are connected hypervisors using the attach entity profile configuration. For details about configuring with attach entity profile, see the examples related to creating VMM domain profiles.

Example:

DHCP Relay Policy for EPG

```
<!-- api/policymgr/mo/.xml -->
<polUni>

POST https://apic-ip-address/api/mo/uni.xml


  <fvTenant name="infra">

    <dhcpRelayP name="DhcpRelayP"  owner="tenant">
      <dhcpRsProv tDn="uni/tn-infra/ap-access/epg-default" addr="10.0.0.1" />
    </dhcpRelayP>

    <fvBD name="default">
      <dhcpLbl name="DhcpRelayP" owner="tenant"/>
    </fvBD>

  </fvTenant>
</polUni>
```

Configuring a DNS Service Policy

A DNS policy is required to connect to external servers, for example AAA, RADIUS, vCenter, and services by hostname. A DNS service policy is a shared policy, so any tenant and VRF that uses this service must be configured with the specific DNS profile label. To configure a DNS policy for the ACI fabric, you must complete the following tasks:

- Ensure that the management EPG is configured for the DNS policy, otherwise this policy will not take into effect on the switches.
- Create a DNS profile (default) that contains the information about DNS providers and DNS domains.

- Associate the DNS profile (default or another DNS profile) name to a DNS label under the required tenant.

It is possible to configure a per-tenant, per-VRF DNS profile configuration. Additional DNS profiles can be created and applied to specific VRFs of specific tenants using the appropriate DNS label. For example, if you create a DNS profile with a name of acme, you can add a DNS label of acme to the appropriate **Networking** > **VRF** policy configuration in the tenants configuration.

Configuring External Destinations with an In-Band DNS Service Policy

Configure the external destinations for the services as follows:

Source	In-Band Management	Out-of-Band Management	External Server Location
APIC	IP address or Fully Qualified domain name (FQDN)	IP address or FQDN	Anywhere
Leaf switches	IP address	IP address or FQDN **Note** The DNS policy must specify the out-of-band management EPG for reachability of the DNS server.	Anywhere
Spine switches	IP address	IP address or FQDN **Note** The DNS policy must specify the out-of-band management EPG for reachability of the DNS server.	Directly connected to a leaf switch

The following is a list of external servers:

- Call Home SMTP server
- Syslog server
- SNMP Trap destination
- Statistics Export destination
- Configuration Export destination
- Techsupport Export destination
- Core Export destination

The recommended guidelines are as follows:

- The external servers must be atatched to the leaf access ports.
- Use in-band connectivity for the leaf switches to avoid extra cabling for the management port.
- Use out-of-band management connectivity for the spine switches. Connect this out-of-band network for spine switches to one of the leaf ports with in-band management virtual routing and forwarding (VRF) so that the spine switches and the leaf switches can reach the same set of external servers.
- Use IP addresses for the external servers.

Policy for Priority of IPv4 or IPv6 in a DNS Profile

The DNS profile supports version preference choices between IPv4 and IPv6. Using the user interface, you can enable your preference. IPv4 is the default.

The following is an example of a policy based configuration using Postman REST API:

```
<?xml version="1.0" encoding="UTF-8”?>
<!— api/node/mo/uni/fabric/dnsp-default.xml —>
<dnsProfile dn="uni/fabric/dnsp-default" IPVerPreference="IPv6" childAction="" descr="" >
</dnsProfile>
```

The gai.conf settings control destination address selection. The file has a label table, precedence table, and an IPv4 scopes table. The changes for prioritizing IPv4 or IPv6 over the other need to go into the precedence table entries. Given below are sample contents of the standard file as it is used in Linux systems for many flavors. A single line of precedence label in the file overrides any default settings.

The following is an example of a gai.conf to prioritize IPv4 over IPv6:

```
# Generated by APIC
label ::1/128       0
label ::/0          1
label 2002::/16     2
label ::/96         3
label ::ffff:0:0/96 4
precedence  ::1/128       50
precedence ::/0           40
precedence 2002::/16      30
precedence ::/96          20
# For APICs prefering IPv4 connections, change the value to 100.
precedence ::ffff:0:0/96  10
```

Dual Stack IPv4 and IPv6 DNS Servers

DNS servers have primary DNS records which can be A records (IPV4) or AAAA records (IPV6). Both A and AAAA records associate domain name with a specific IP address (IPv4 or IPv6).

The ACI fabric can be configured to use reputable public DNS servers that run on IPv4. These servers are able to resolve and respond with A record (IPv4) or AAAA record (IPv6).

In a pure IPv6 environment, the system administrators must use IPv6 DNS servers. The IPv6 DNS servers are enabled by adding them to /etc/resolv.conf.

A more common environment is to have dual-stack IPv4 and IPv6 DNS servers. In the dual-stack case, both IPv4 and IPv6 name servers are listed in /etc/resolv.conf. However, in a dual-stack environment, simply appending the IPv6 DNS servers to the list may cause a large delay in DNS resolutions. This is because the IPv6 protocol takes precedence by default, and it is unable to connect to the IPv4 DNS servers (if they are listed first in /etc/resolv.conf). The solution is to list IPv6 DNS servers ahead of IPv4 DNS servers. Also add “options single-request-reopen” to enable the same socket to be used for both IPv4 and IPv6 lookups.

Here is an example of resolv.conf in dual-stack IPv4 and IPv6 DNS servers where the IPv6 DNS servers are listed first. Also note the “single-request-reopen” option:

```
options single-request-reopen
nameserver 2001:4860:4680::8888
nameserver 2001:4860:4680::8844
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Dual-Stack IPv4 and IPv6 Environment

If the management network in the ACI fabric supports both IPv4 and IPv6, the Linux system application (glibc) will use the IPv6 network by default because getaddrinfo() will return IPv6 first.

Under certain conditions however, an IPv4 address may be preferred over an IPv6 address. The Linux IPv6 stack has a feature which allows an IPv4 address mapped as an IPv6 address using IPv6 mapped IPv4 address (::ffff/96). This allows an IPv6 capable application to use only a single socket to accept or connect both IPv4 and IPv6. This is controlled by the glibc IPv6 selection preference for getaddrinfo() in /etc/gai.conf.

In order to allow glibc to return multiple addresses when using /etc/hosts, “multi on” should be added to the /etc/hosts file. Otherwise, it may return only the first match.

If an application is not aware whether both IPv4 and IPv6 exist, it may not perform fallback attempts using different address families. Such applications may require a fallback implementation.

Configuring a DNS Service Policy to Connect with DNS Providers Using the REST API

Before You Begin

Make sure that Layer 2 or Layer 3 management connectivity is configured.

Procedure

Step 1 Configure the DNS service policy.

Example:

```
POST URL :
https://apic-IP-address/api/node/mo/uni/fabric.xml

<dnsProfile name="default">

    <dnsProv addr="172.21.157.5" preferred="yes"/>
    <dnsProv addr="172.21.157.6"/>

    <dnsDomain name="cisco.com" isDefault="yes"/>

    <dnsRsProfileToEpg tDn="uni/tn-mgmt/mgmtp-default/oob-default"/>

</dnsProfile>
```

Step 2 Configure the DNS label under the out-of-band management tenant.

Example:

```
POST URL: https://apic-IP-address/api/node/mo/uni/tn-mgmt/ctx-oob.xml
<dnsLbl name="default" tag="yellow-green"/>
```

Verifying that the DNS Profile is Configured and Applied to the Fabric Controller Switches Using the CLI

Procedure

Step 1 Verify the configuration for the default DNS profile.

Example:

```
admin@apic1:~> cd /aci/fabric/fabric-policies/global-policies/dns-profiles/default
admin@apic1:default> cat summary
# dns-profile
name : default
description : added via CLI by tdeleon@cisco.com
ownerkey :
ownertag :

dns-providers:
address preferred
-------------- ---------
10.44.124.122 no
10.70.168.183 no
10.37.87.157 no
10.102.6.247 yes
dns-domains:
name default description
--------- -------- -----------
cisco.com yes
management-epg : tenants/mgmt/node-management-epgs/default/out-of-band/default
```

Step 2 Verify the configurations for the DNS labels.

Example:

```
admin@apic1:default> cd
/aci/tenants/mgmt/networking/private-networks/oob/dns-profile-labels/default
admin@apic1:default> cat summary
# dns-lbl
name : default
description :
ownerkey :
ownertag :
tag : yellow-green
```

Step 3 Verify that the applied configuration is operating on the fabric controllers.

Example:

```
admin@apic1:~> cat /etc/resolv.conf
# Generated by IFC
search cisco.com
nameserver 10.102.6.247
nameserver 10.44.124.122
nameserver 10.37.87.157
nameserver 10.70.168.183
admin@apic1:~> ping www.cisco.com
PING origin-www.cisco.com (72.163.4.161) 56(84) bytes of data.
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=1 ttl=238 time=35.4 ms
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=2 ttl=238 time=29.0 ms
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=3 ttl=238 time=29.2 ms
```

Step 4 Verify that the applied configuration is operating on the leaf and spine switches.

Example:

```
leaf1# cat /etc/resolv.conf
search cisco.com
nameserver 10.102.6.247
nameserver 10.70.168.183
nameserver 10.44.124.122
nameserver 10.37.87.157
leaf1# cat /etc/dcos_resolv.conf
# DNS enabled
leaf1# ping www.cisco.com
PING origin-www.cisco.com (72.163.4.161): 56 data bytes
64 bytes from 72.163.4.161: icmp_seq=0 ttl=238 time=29.255 ms
64 bytes from 72.163.4.161: icmp_seq=1 ttl=238 time=29.212 ms
64 bytes from 72.163.4.161: icmp_seq=2 ttl=238 time=29.343 ms
```

Configuring External Connectivity for Tenants

Before you can distribute the static route to the other leaf switches on the Application Centric Infrastructure (ACI) fabric, a multiprotocol BGP (MP-BGP) process must first be operating, and the spine switches must be configured as BGP route reflectors.

To integrate the ACI fabric into an external routed network, you can configure Open Shortest Path First (OSPF) for management tenant Layer 3 connectivity.

Configuring an MP-BGP Route Reflector Using the REST API

Procedure

Step 1 Mark the spine switches as route reflectors.

Example:

```
POST https://apic-ip-address/api/policymgr/mo/uni/fabric.xml

<bgpInstPol name="default">
  <bgpAsP asn="1" />
  <bgpRRP>
     <bgpRRNodePEp id=“<spine_id1>”/>
     <bgpRRNodePEp id=“<spine_id2>”/>
  </bgpRRP>
</bgpInstPol>
```

Step 2 Set up the pod selector using the following post.

Example:

For the FuncP setup—

```
POST https://apic-ip-address/api/policymgr/mo/uni.xml

<fabricFuncP>
  <fabricPodPGrp name="bgpRRPodGrp”>
    <fabricRsPodPGrpBGPRRP tnBgpInstPolName="default" />
  </fabricPodPGrp>
</fabricFuncP>
```

Example:
For the PodP setup—

```
POST https://apic-ip-address/api/policymgr/mo/uni.xml

<fabricPodP name="default">
  <fabricPodS name="default" type="ALL">
    <fabricRsPodPGrp tDn="uni/fabric/funcprof/podpgrp-bgpRRPodGrp"/>
  </fabricPodS>
</fabricPodP>
```

Verifying the MP-BGP Route Reflector Configuration

Procedure

Step 1 Verify the configuration by performing the following actions:

a) Use secure shell (SSH) to log in as an administrator to each leaf switch as required.
b) Enter the **show processes | grep bgp** command to verify the state is S.
   If the state is NR (not running), the configuration was not successful.

Step 2 Verify that the autonomous system number is configured in the spine switches by performing the following actions:

a) Use the SSH to log in as an administrator to each spine switch as required.
b) Execute the following commands from the shell window

   Example:
   cd /mit/sys/bgp/inst

   Example:
   grep asn summary

The configured autonomous system number must be displayed. If the autonomous system number value displays as 0, the configuration was not successful.

Creating OSPF External Routed Network for Management Tenant Using REST API

- You must verify that the router ID and the logical interface profile IP address are different and do not overlap.
- The following steps are for creating an OSPF external routed network for a management tenant. To create an OSPF external routed network for a tenant, you must choose a tenant and create a VRF for the tenant.

- For more details, see *Cisco APIC and Transit Routing*.

Procedure

Create an OSPF external routed network for management tenant.

Example:

```
POST: https://apic-ip-address/api/mo/uni/tn-mgmt.xml

<fvTenant name="mgmt">
   <fvBD name="bd1">
      <fvRsBDToOut tnL3extOutName="RtdOut" />
      <fvSubnet ip="1.1.1.1/16" />
      <fvSubnet ip="1.2.1.1/16" />
      <fvSubnet ip="40.1.1.1/24" scope="public" />
      <fvRsCtx tnFvCtxName="inb" />
   </fvBD>
   <fvCtx name="inb" />

   <l3extOut name="RtdOut">
      <l3extRsL3DomAtt tDn="uni/l3dom-extdom"/>
      <l3extInstP name="extMgmt">
      </l3extInstP>
      <l3extLNodeP name="borderLeaf">
         <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="10.10.10.10"/>
         <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-102" rtrId="10.10.10.11"/>
         <l3extLIfP name='portProfile'>
           <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/40]"
ifInstT='l3-port' addr="192.168.62.1/24"/>
           <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-102/pathep-[eth1/40]"
ifInstT='l3-port' addr="192.168.62.5/24"/>
            <ospfIfP/>
                                </l3extLIfP>
      </l3extLNodeP>
      <l3extRsEctx tnFvCtxName="inb"/>
      <ospfExtP areaId="57" />
   </l3extOut>
</fvTenant>
```

Deploying an Application Policy

Three-Tier Application Deployment

A filter specifies the data protocols to be allowed or denied by a contract that contains the filter. A contract can contain multiple subjects. A subject can be used to realize uni- or bidirectional filters. A unidirectional filter is a filter that is used in one direction, either from consumer-to-provider (IN) or from provider-to-consumer (OUT) filter. A bidirectional filter is the same filter that is used in both directions. It is not reflexive.

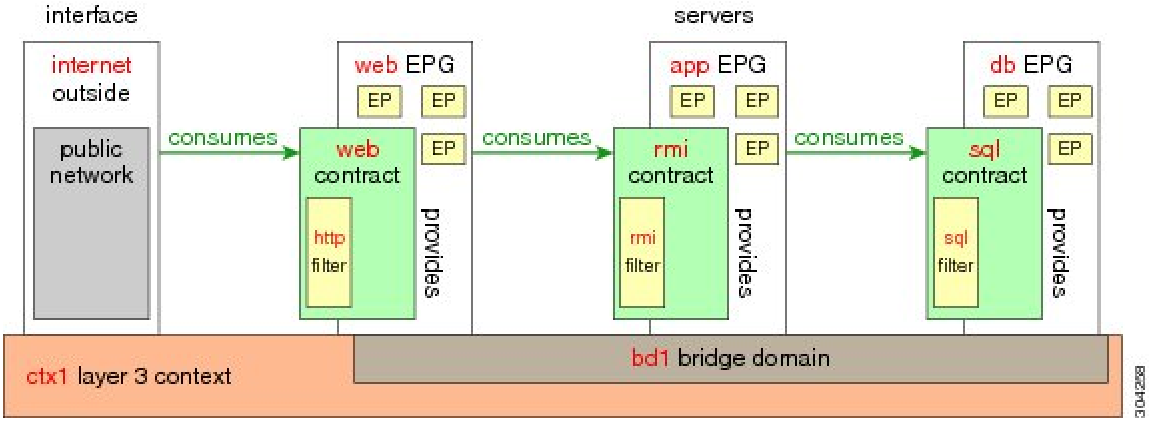
Contracts are policies that enable inter-End Point Group (inter-EPG) communication. These policies are the rules that specify communication between application tiers. If no contract is attached to the EPG, inter-EPG communication is disabled by default. No contract is required for intra-EPG communication because intra-EPG communication is always allowed.

Application profiles enable you to model application requirements that the APIC then automatically renders in the network and data center infrastructure. The application profiles enable administrators to approach the resource pool in terms of applications rather than infrastructure building blocks. The application profile is a container that holds EPGs that are logically related to one another. EPGs can communicate with other EPGs in the same application profile and with EPGs in other application profiles.

To deploy an application policy, you must create the required application profiles, filters, and contracts. Typically, the APIC fabric hosts a three-tier application within a tenant network. In this example, the application is implemented by using three servers (a web server, an application server, and a database server). See the following figure for an example of a three-tier application.

The web server has the HTTP filter, the application server has the Remote Method Invocation (RMI) filter, and the database server has the Structured Query Language (SQL) filter. The application server consumes the SQL contract to communicate with the database server. The web server consumes the RMI contract to communicate with the application server. The traffic enters from the web server and communicates with the application server. The application server then communicates with the database server, and the traffic can also communicate externally.

Figure 3: Three-Tier Application Diagram



Parameters to Create a Filter for http

The parameters to create a filter for http in this example is as follows:

Parameter Name	Filter for http
Name	http
Number of Entries	2
Entry Name	Dport-80 Dport-443
Ethertype	IP
Protocol	tcp tcp
Destination Port	http https

Parameters to Create Filters for rmi and sql

The parameters to create filters for rmi and sql in this example are as follows:

Parameter Name	Filter for rmi	Filter for sql
Name	rmi	sql
Number of Entries	1	1
Entry Name	Dport-1099	Dport-1521
Ethertype	IP	IP
Protocol	tcp	tcp
Destination Port	1099	1521

Example Application Profile Database

The application profile database in this example is as follows:

EPG	Provided Contracts	Consumed Contracts
web	web	rmi
app	rmi	sql
db	sql	--

Deploying an Application Profile Using the REST API

The port the EPG uses must belong to one of the VM Managers (VMM) or physical domains associated with the EPG.

Procedure

Step 1 Send this HTTP POST message to deploy the application using the XML API.

Example:

```
POST https://apic-ip-address/api/mo/uni/tn-ExampleCorp.xml
```

Step 2 Include this XML structure in the body of the POST message.

Example:

```
<fvTenant name="ExampleCorp">

    <fvAp name="OnlineStore">
        <fvAEPg name="web">
            <fvRsBd tnFvBDName="bd1"/>
            <fvRsCons tnVzBrCPName="rmi"/>
            <fvRsProv tnVzBrCPName="web"/>
            <fvRsDomAtt tDn="uni/vmmp-VMware/dom-datacenter"delimiter=@/>
        </fvAEPg>

        <fvAEPg name="db">
            <fvRsBd tnFvBDName="bd1"/>
            <fvRsProv tnVzBrCPName="sql"/>
            <fvRsDomAtt tDn="uni/vmmp-VMware/dom-datacenter"/>
        </fvAEPg>

        <fvAEPg name="app">
            <fvRsBd tnFvBDName="bd1"/>
            <fvRsProv tnVzBrCPName="rmi"/>
            <fvRsCons tnVzBrCPName="sql"/>
            <fvRsDomAtt tDn="uni/vmmp-VMware/dom-datacenter"/>
        </fvAEPg>
    </fvAp>

<vzFilter name="http" >
<vzEntry dFromPort="80" name="DPort-80" prot="tcp" etherT="ip"/>
<vzEntry dFromPort="443" name="DPort-443" prot="tcp" etherT="ip"/>
</vzFilter>
<vzFilter name="rmi" >
<vzEntry dFromPort="1099" name="DPort-1099" prot="tcp" etherT="ip"/>
</vzFilter>
<vzFilter name="sql">
<vzEntry dFromPort="1521" name="DPort-1521" prot="tcp" etherT="ip"/>
</vzFilter>
    <vzBrCP name="web">
        <vzSubj name="web">
            <vzRsSubjFiltAtt tnVzFilterName="http"/>
        </vzSubj>
    </vzBrCP>

    <vzBrCP name="rmi">
        <vzSubj name="rmi">
            <vzRsSubjFiltAtt tnVzFilterName="rmi"/>
        </vzSubj>
    </vzBrCP>

    <vzBrCP name="sql">
        <vzSubj name="sql">
           <vzRsSubjFiltAtt tnVzFilterName="sql"/>
        </vzSubj>
    </vzBrCP>
</fvTenant>
```

In the string fvRsDomAtt tDn="uni/vmmp-VMware/dom-datacenter"delimiter=@/, **delimiter=@** is optional. If you do not enter a delimiter, the system will use the default | delimiter.

In the XML structure, the first line modifies, or creates if necessary, the tenant named ExampleCorp.

```
<fvTenant name="ExampleCorp">
```

This line creates an application network profile named OnlineStore.

```
<fvAp name="OnlineStore">
```

The elements within the application network profile create three endpoint groups, one for each of the three servers. The following lines create an endpoint group named web and associate it with an existing bridge domain named bd1. This endpoint group is a consumer, or destination, of the traffic allowed by the binary contract named rmi and is a provider, or source, of the traffic allowed by the binary contract named web. The endpoint group is associated with the VMM domain named datacenter.

```
<fvAEPg name="web">
    <fvRsBd tnFvBDName="bd1"/>
    <fvRsCons tnVzBrCPName="rmi"/>
    <fvRsProv tnVzBrCPName="web"/>
    <fvRsDomAtt tDn="uni/vmmp-VMware/dom-datacenter"/>
</fvAEPg>
```

The remaining two endpoint groups, for the application server and the database server, are created in a similar way.

The following lines define a traffic filter named http that specifies TCP traffic of types HTTP (port 80) and HTTPS (port 443).

```
<vzFilter name="http" >
<vzEntry dFromPort="80" name="DPort-80" prot="tcp" etherT="ip"/>
<vzEntry dFromPort="443" name="DPort-443" prot="tcp" etherT="ip"/>
</vzFilter>
```

The remaining two filters, for application data and database (sql) data, are created in a similar way.

The following lines create a binary contract named web that incorporates the filter named http:

```
<vzBrCP name="web">
    <vzSubj name="web">
        <vzRsSubjFiltAtt tnVzFilterName="http"/>
    </vzSubj>
</vzBrCP>
```

The remaining two contracts, for rmi and sql data protocols, are created in a similar way.

The final line closes the structure:

```
</fvTenant>
```